Monte-Carlo MagicPIG : An asymptotically consistent estimator for attention head outputs

Swann Bessa

April 24, 2025

Abstract

MagicPIG is a method that approximates the output of an attention head, and yields impressive accuracy on several benchmarks with much less computational cost than full attention. It is based on LSH (Locality-Sensitive Hashing), and computes the attention output by sampling and reweighing a subset of key indices. To do so, the original paper combines two theoretically grounded statistical methods : Self-Normalized Importance Sampling and LSH Sampling. However, we argue that the current combination of those two methods in the MagicPIG estimator could be improved to reinforce theoretical guarantees, which could result in more efficient sampling. Moreover, we find out that the sampling method used in MagicPIG is in fact highly deterministic, and almost surely selects all keys which exhibit high cosine similarity with the query. This makes the method surprisingly similar to Top-K attention. From these observations, we propose an extension of the method called Monte-Carlo MagicPIG, which averages attention output estimations over multiple subsets obtained from independent LSH Sampling procedures. We prove theoretical guarantees for the resulting estimator X_{MCmPIG} and hypothesize that it could lead to a better trade-off between accuracy, computational cost and memory overhead.

1 Problem definition

Given sets of keys $K = [k_1, k_2, \ldots, k_n]^\top \in \mathbb{R}^{n\times d}$, values $V = [v_1, v_2, \ldots, v_n]^\top \in \mathbb{R}^{n\times p}$, and a query $q \in \mathbb{R}^d$, we aim to approximate the output of an attention head with a trade-off between accuracy, computational cost and memory overhead:

$$\text{Attention}(q, K, V) = \sum_{i=1}^{n} w_i v_i, \tag{1}$$

where the attention weights w_i are defined as:

$$w_i = \frac{\exp\left(q^\top k_i/\sqrt{d}\right)}{\sum_{j=1}^{n} \exp\left(q^\top k_j/\sqrt{d}\right)}. \tag{2}$$

2 The MagicPIG approach

2.1 Estimation of the attention head output

To approximate this attention head output, the core idea of MagicPIG [2] is to combine self-normalized importance sampling [4] and LSH sampling [6]. Let us first focus on self-normalized importance sampling given an unnormalized proposal distribution μ.

Definition 1. Given an unnormalized proposal distribution μ, and i.i.d. samples i_j from that distribution, the self-normalized Importance Sampling estimator for the attention head output is :

$$X^{\text{IS}} = \frac{1}{\widetilde{Z}} \sum_{j=1}^{B} \frac{\widetilde{w}_{i_j}}{\mu_{i_j}} v_{i_j},$$

where the normalizing constant $\widetilde{Z}$ is given by:

$$\widetilde{Z} = \sum_{j=1}^{B} \frac{\widetilde{w}_{i_j}}{\mu_{i_j}},$$

and $\widetilde{w}_i = e^{q^\top k_i/\sqrt{d}}$ is the weight associated with the sampled indices i_j.

[4] derives asymptotic guarantees for the Self-Normalized Importance Sampling estimator, hence the following theorem :

Theorem 1 (Self-Normalized Importance Sampling). The Self-Normalized Importance Sampling estimator for the attention head output is asymptotically consistent :
$\mathbb{P}\left[\lim_{B\to\infty} X^{\text{IS}} = \text{Attention}(q, K, V)\right] = 1.$

MagicPIG then highlights that the variance of the self-normalized IS estimator can best be reduced with a proposal distribution μ such that μ_i is monotonic in $q^\top k_i$. Thus, LSH as a sampler [1] [6] is then used with SimHash [5], a LSH family that preserves cosine similarity : this method allows to sample a subset of keys S such that each key k belongs to S with a probability monotonic in its cosine similarity. In its original form proposed in [6], LSH as a sampler is proven to be an unbiased estimator. Directly applying LSH sampling to the estimation of the attention head output yields the following result :

Lemma 1 (LSH Sampling). $X_{IS} = \sum_{i=1}^{n} \frac{w_i}{u_i} v_i \mathbb{1}\{i \in S\}$, where $\mathbb{P}(i \in S) = u_i$, is an unbiased estimator of Attention(q, K, V).

Proof. The linearity of the expectation directly yields $E(X_{IS}) = \sum_{i=1}^{n} w_i v_i$, which corresponds exactly to Attention(q,K,V).

□

However, this LSH sampling cannot be applied directly in our case since the normalized weights w_i are unknown. Combining self-normalized importance sampling and LSH as a sampler, the authors thus propose the following estimator for the attention head output :

$$X_{mPIG} = \frac{\sum_{i=1}^{n} \frac{\widetilde{w}_i}{u_i} v_i \mathbb{1}_{i \in S}}{\sum_{i=1}^{n} \frac{\widetilde{w}_i}{u_i} \mathbb{1}_{i \in S}} = \frac{\sum_{i \in S} \frac{\widetilde{w}_i}{u_i} v_i}{\sum_{i \in S} \frac{\widetilde{w}_i}{u_i}} \tag{3}$$

2.2 Limitations of the approach

As mentioned in the original paper, the estimator X_{mPIG} is an approximation. The use of LSH sampling implies that samples are drawn **without replacement** in the MagicPIG method. This yields two main limitations in the context of estimating the attention head output :

- Since self-normalized importance sampling requires sampling **with replacement** for the proof of consistency [4], there is, to the best of our knowledge, no convergence guarantee of the method.
- Sampling without replacement implies that there are **no duplicate samples**. However, in our case, duplicate samples have the computational benefit of providing multiple samples while requiring only one dot product $q^\top k_i$ in the final formula. In the original paper, massive duplication explains why Oracle Sampling has a low $Cost_2$ for a high accuracy. This lack of duplicates limits the efficiency of MagicPIG's sampling method in our specific context.

Moreover, since weights $\widetilde{w}_i$ are not normalized, Lemma 1 cannot be applied. To our knowledge, there is thus no theoretical guarantees of either unbiasedness or consistency.

2.3 MagicPIG and top-K attention : two surprisingly similar methods

In this subsection, we hypothesize that the impressive experimental results of LSH Sampling are in fact due to a high resemblance with top-K attention [3], rather than an effective sampling procedure. Indeed, the analysis of the sampling probabilities of LSH Sampling shows that its behaviour is in fact highly deterministic, thus yielding attention head outputs that are close to those returned by a Top-K attention method.

Result 1. Given a key k_i and a query q, denoting $\theta \in (-\pi, \pi)$ the angle between k_i and q, and (K, L) the LSH sampling hyper-parameters, we have the following result :
$\mathbb{P}(i \in S) = 1 - \left(1 - (1 - |\frac{\theta}{\pi}|)^K\right)^L - L \cdot (1 - |\frac{\theta}{\pi}|)^K \cdot \left(1 - (1 - |\frac{\theta}{\pi}|)^K\right)^{(L-1)}$

The following figure plots the function yielded by Result 1 : the probability of sampling an indice i (and its corresponding key k_i) as a function of the angle between q and k_i, with typical hyper-parameters $(K, L) = (10, 220)$:

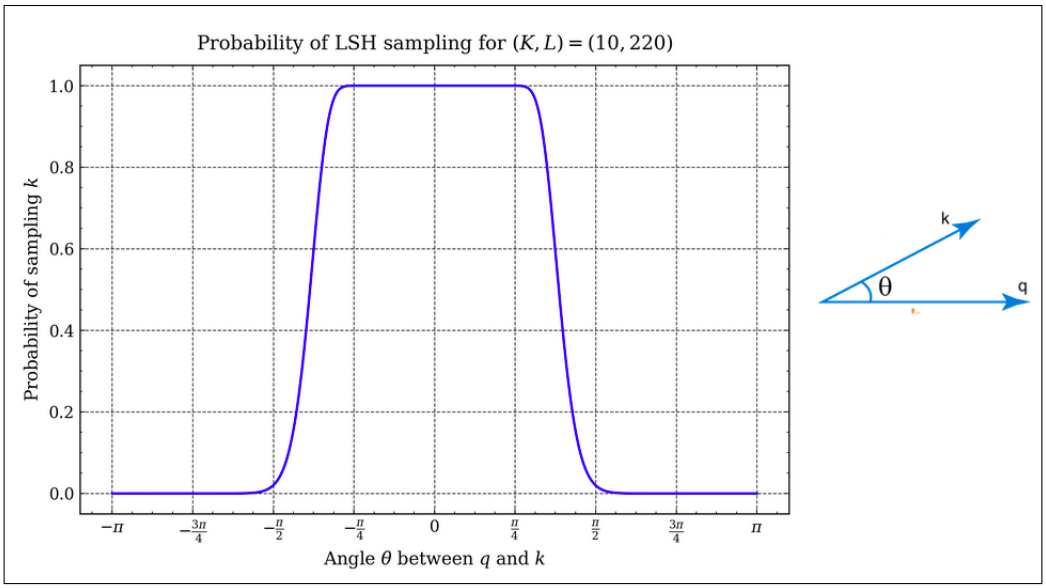

Figure 1: Probability of sampling a key k given its angle θ with the query q. The figure shows only one set of hyper-parameters (K, L) used in the original paper. The appendix shows that the other sets of hyper-parameters used in the paper exhibit very similar behavior.

The figure shows an interesting saturation phenomenon of the sampling distribution. For almost 85% of the angles, the sampling procedure is in fact deterministic. Keys with angles θ in $[-\frac{\pi}{3}, \frac{\pi}{3}]$ are almost surely sampled, while keys with $|\theta| > \frac{\pi}{2}$ are almost surely not sampled. This observation is summarized in Figure 2.

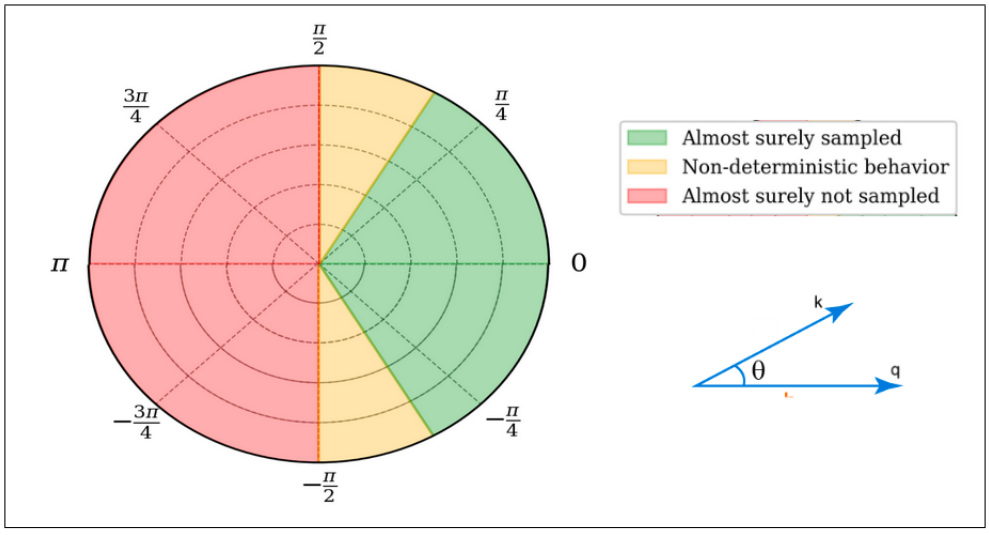

Figure 2: Sampling behavior for keys depending on their angle θ with the query q. Keys with high cosine similarity with q are almost surely sampled, while keys with low cosine similarity are almost surely not sampled

Therefore, all the keys with top cosine-similarity are systematically sampled, while the keys with negative cosine-similarity are almost surely not sampled. This makes the behavior

of the MagicPIG method close to a Top-K method (with an additional reweighting).

Thus, we hypothesize that the impressive experimental results of MagicPIG are more likely due to an efficient nearest neighbor algorithm on cosine similarity rather than an efficient sampling. **We argue that the strength of MagicPIG lies in its elegant nearest-neighbor computation, which naturally addresses the OOD issue between keys and queries raised in [3] by leveraging Locality-Sensitive Hashing.**

3 Proposed approach : Monte-Carlo MagicPIG

Aiming to reinforce the theoretical guarantees on X_{mPIG} and better exploit the stochastic behavior of LSH Sampling [6], we extend the approach by proposing a provably consistent estimator X_{MCmPIG} which leverages Monte-Carlo sampling.

Instead of performing LSH sampling on only one set of hash tables with hyper-parameters (K, L), we propose to perform LSH sampling on M smaller sets of hash tables $(K_h, L_h)_{h=1,...,M}$, each yielding a subset of keys S_h. The estimations of the attention head output for each of these subsets are then averaged into one single estimator, provided in the following formula :

$$X_{MCmPIG} = \frac{\frac{1}{M}\sum_{h=1}^{M}\sum_{i=1}^{n}\frac{\widetilde{w}_i}{u_i}v_i\mathbb{1}_{i\in S_h}}{\frac{1}{M}\sum_{h=1}^{M}\sum_{i=1}^{n}\frac{\widetilde{w}_i}{u_i}\mathbb{1}_{i\in S_h}} = \frac{\sum_{h=1}^{M}\sum_{i\in S_h}\frac{\widetilde{w}_i}{u_i}v_i}{\sum_{h=1}^{M}\sum_{i\in S_h}\frac{\widetilde{w}_i}{u_i}} \tag{4}$$

While several sets of hash tables can result in significantly more memory consumption, we argue that such a concern can be avoided. Since the (K, L) values in the paper can go up to $(10, 220)$, we propose to use multiple **much smaller** sets of hash tables $(K_h, L_h)_{h=1,...,M}$ ($\forall h \in \{1, ..., M\}, K_h L_h \ll KL$), such that the total CPU memory consumption is comparable ($\sum_{h=1}^{M} K_h L_h \approx KL$). We hypothesize that using this method, a better trade-off between accuracy and computational cost can be achieved. As proven before, MagicPIG's (K, L) hyper-parameters mostly showcase deterministic behavior. The appendix shows that much smaller values of these hyper-parameters exhibit more stochastic behavior. **We thus believe that much smaller values of (K_h, L_h) can allow to better exploit the sampling view of Locality-Sensitive Hashing.**

Moreover, since the cost of retrieving S is not a bottleneck in the original paper ($Cost_1 \approx 0$), we believe that the estimation gain from our estimator can compensate for a potential memory overhead.

3.1 Consistency of the estimator

A proof of consistency for our estimator can be derived with elementary statistics. We assume that all the hash tables have the same parameters : (K_h, L_h) are the same for all h. Then the following theorem can be derived :

Theorem 2. X_{MCmPIG} is an asymptotically consistent estimator for the attention head output : $\mathbb{P}\left[\lim_{M\to\infty} X^{\mathrm{MCmPIG}} = \mathrm{Attention}(q, K, V)\right] = 1$

Proof. In the following proof we denote : $Z = \sum_{i=1}^{n} \widetilde{w}_i$. Given that all sets of hash tables have the same hyper-parameters (K_h, L_h), the probability laws don't depend on the parameter h: $\forall h \in \{1, ..., M\}, \forall i \in \{1, ..., n\}, \mathbb{P}(i \in S_h) = u_i$

Let us recall the formula : $X_{MCmPIG} = \frac{\frac{1}{M}\sum_{h=1}^{M}\sum_{i=1}^{n}\frac{\widetilde{w}_i}{u_i}v_i\mathbb{1}_{i\in S_h}}{\frac{1}{M}\sum_{h=1}^{M}\sum_{i=1}^{n}\frac{\widetilde{w}_i}{u_i}\mathbb{1}_{i\in S_h}}$

The numerator of X_{MCmPIG} is:

$$\frac{1}{M}\sum_{h=1}^{M}\sum_{i=1}^{n}\frac{\widetilde{w}_i}{u_i}v_i\mathbb{1}_{i\in S_h}.$$

For a single h and i, the expectation over the randomness of S_h is:

$$\mathbb{E}\left[\frac{\widetilde{w}_i}{u_i}v_i\mathbb{1}_{i\in S_h}\right] = \frac{\widetilde{w}_i}{u_i}v_i \cdot u_i = \widetilde{w}_i v_i$$

Summing over all i, the expected value of one summand of the numerator is:

$$\mathbb{E}\left[\sum_{i=1}^{n}\frac{\widetilde{w}_i}{u_i}v_i\mathbb{1}_{i\in S_h}\right] = \sum_{i=1}^{n}\widetilde{w}_i v_i$$

Similarly, the denominator is:

$$\frac{1}{M}\sum_{h=1}^{M}\sum_{i=1}^{n}\frac{\widetilde{w}_i}{u_i}\mathbb{1}_{i\in S_h}$$

For a single h and i, the expectation over the randomness of S_h is:

$$\mathbb{E}\left[\frac{\widetilde{w}_i}{u_i}\mathbb{1}_{i\in S_h}\right] = \frac{\widetilde{w}_i}{u_i} \cdot u_i = \widetilde{w}_i$$

Summing over all i, the expected value of one summand of the denominator is:

$$\mathbb{E}\left[\sum_{i=1}^{n}\frac{\widetilde{w}_i}{u_i}\mathbb{1}_{i\in S_h}\right] = \sum_{i=1}^{n}\widetilde{w}_i = Z$$

Given that the hash tables $(K_h, L_h)_{i=1,...,m}$ are built independently, the subsets S_h are i.i.d. samples. Therefore, by the **law of large numbers (LLN)**, as $M \to \infty$, both the numerator and denominator converge almost surely to the expectations previously computed :

$$\frac{1}{M}\sum_{h=1}^{M}\sum_{i=1}^{n}\frac{\widetilde{w}_i}{u_i}v_i\mathbb{1}_{i\in S_h} \xrightarrow{a.s.} \sum_{i=1}^{n}\widetilde{w}_i v_i,$$

and :

$$\frac{1}{M}\sum_{h=1}^{M}\sum_{i=1}^{n}\frac{\widetilde{w}_i}{u_i}\mathbb{1}_{i\in S_h} \xrightarrow{a.s.} \sum_{i=1}^{n}\widetilde{w}_i = Z.$$

Substituting these results into X_{MCmPIG}, we have :

$$X_{MCmPIG} \xrightarrow{a.s.} \frac{\sum_{i=1}^{n} \widetilde{w}_i v_i}{\sum_{i=1}^{n} \widetilde{w}_i}.$$

Since $w_i = \frac{\widetilde{w}_i}{Z}$, this simplifies to :

$$X_{MCmPIG} \xrightarrow{a.s.} \sum_{i=1}^{n} w_i v_i = \text{Attention}(q, K, V).$$

□

3.2 An estimator that leverages duplication

The original paper mentions that massive sample duplication is witnessed during Oracle Sampling. As previously stated, this phenomenon allows the Oracle Sampling estimator to converge towards the attention head output while not requiring too many dot-products $q^T k_i$ to compute in the second phase (low $Cost_2$).

While Oracle Sampling is not computationally efficient (since the full attention has to be computed in the first phase, hence the high $Cost_1$), the low value of $Cost_2$ offers an interesting insight into how an efficient sampler can be obtained thanks to duplication. In addition to providing consistency guarantees, X_{MCmPIG} leverages this duplication phenomenon as shown by the following rewriting :

$$X_{MCmPIG} = \frac{\sum_{i=1}^{n} \frac{\widetilde{w}_i}{u_i} v_i \sum_{h=1}^{M} \mathbb{1}_{i \in S_h}}{\sum_{i=1}^{n} \frac{\widetilde{w}_i}{u_i} \sum_{h=1}^{M} \mathbb{1}_{i \in S_h}}$$

Since our method allows to sample **with replacement**, a same indice i can be sampled in multiple subsets S_h, thus providing more stochastic information while only requiring one dot-product $q^T k_i$ in the final formula.

3.3 Discussion and Conclusion

This approach extends the MagicPIG framework by averaging the subsets S_h from M sets of LSH samplers $(K_h, L_h)_{h=1,\dots,M}$. The intuition behind it is to better exploit the sampling ability of LSH and leverage sample duplication. The studies in the appendix show that very low values of (K_h, L_h) can exhibit stochastic behavior suited to our approach. We thus believe that values above $M = 200$ are realistic using the same CPU memory as MagicPIG. This could allow to come close to the asymptotic regime of Monte-Carlo averaging and thus provide precise estimates of the attention output.

However, in order to attain a realistically appliable method, a careful tuning of the hyper-parameters $(K_h, L_h)_{h=1,\dots,M}$ and M is needed. Another potential optimisation would consist in modifying the number of hash tables collisions needed for retrieval in the LSH sampling. Then, an optimal trade-off between computational cost, memory overhead and accuracy could be found. We believe that such a trade-off could enhance the already impressive performance of MagicPIG.

Appendix : LSH Sampling depending on (K, L)

The sampling probabilities are highly sensitive in the hyper-parameters (K, L). While the sets of hyper-parameters used in the MagicPIG approach showcase highly deterministic behavior, we show in the following figure that much smaller values allow for much more stochastic behavior.

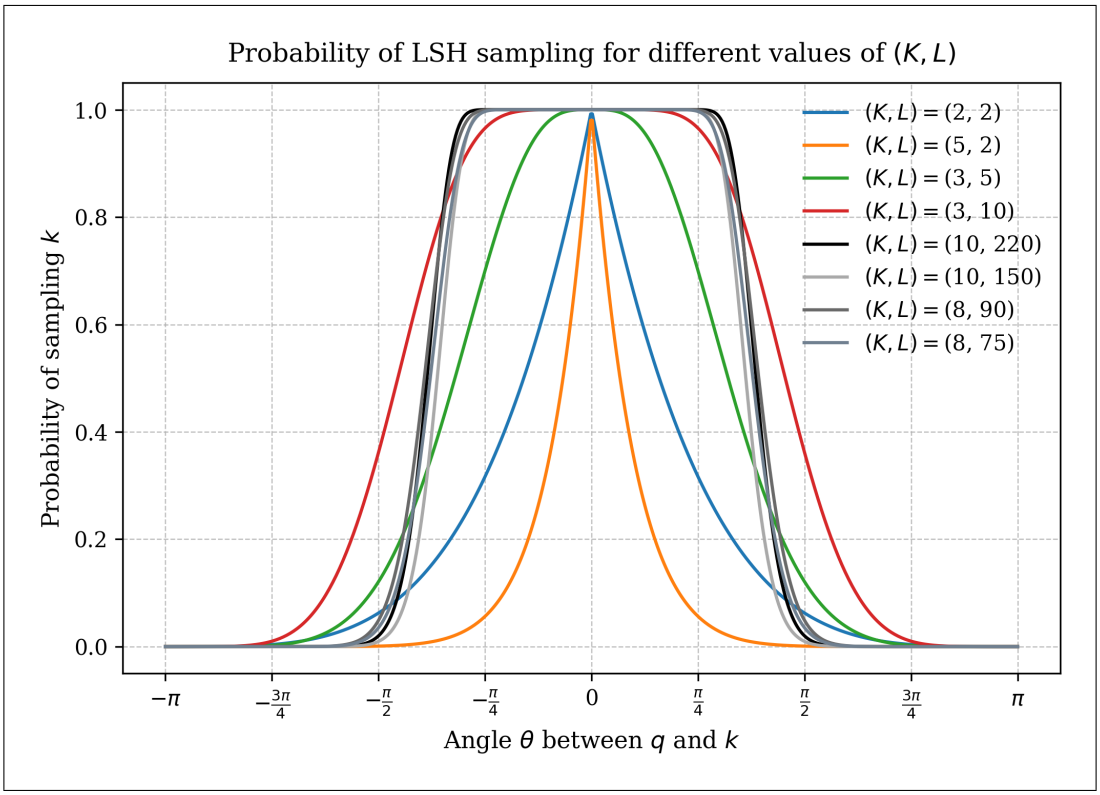

Figure 3: LSH Sampling behavior depending on (K, L). The (K, L) used in MagicPIG are represented in shades of gray/black and exhibit highly deterministic behavior. The other colors represent smaller values of (K, L) showcasing much more stochastic behavior, that could potentially be used for our proposed estimator X_{MCmPIG}.

Interestingly, another hyper-parameter that can be controlled is the number of hash tables collisions between the key and the query for the key required to be sampled. For example, if we only require one collision, which is quite often done in the literature [6], the sampling probability function becomes : $\mathbb{P}(i \in S) = 1 - \left(1 - (1 - |\frac{\theta}{\pi}|)^K\right)^L$

References

[1] Beidi Chen, Tharun Medini, James Farwell, Sameh Gobriel, Charlie Tai, and Anshumali Shrivastava. Slide: In defense of smart algorithms over hardware acceleration for large-scale deep learning systems. In *Proceedings of Machine Learning and Systems*, 2020.

[2] Zhuoming Chen, Ranajoy Sadhukhan, Zihao Ye, Yang Zhou, Jianyu Zhang, Niklas Nolte, Yuandong Tian, Matthijs Douze, Leon Bottou, Zhihao Jia, and Beidi Chen. Magicpig: Lsh sampling for efficient llm generation. *CSAI*, 2024.

[3] Di Liu, Meng Chen, Baotong Lu, Huiqiang Jiang, Zhenhua Han, Qianxi Zhang, Qi Chen, Chengruidong Zhang, Bailu Ding, Kai Zhang, Chen Chen, Fan Yang, Yuqing Yang, and Lili Qiu. Retrievalattention: Accelerating long-context llm inference via vector retrieval. 2024.

[4] Art B. Owen. *Monte Carlo Theory, Methods and Examples.* 2013.

[5] Caitlin Sadowski. Simhash: Hash-based similarity detection. 2007.

[6] R. Spring and A. Shrivastava. A new unbiased and efficient class of lsh-based samplers and estimators for partition function computation in log-linear models. 2017.